

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-202923  
(43)Date of publication of application : 22.07.1994

J1017 U.S. PTO  
09/823755  
04/03/01

(51)Int.Cl.

G06F 12/00

(21)Application number : 04-360787

(71)Applicant : OMRON CORP

(22)Date of filing : 29.12.1992

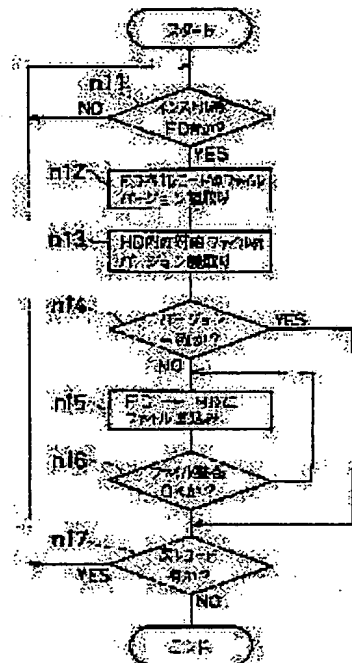
(72)Inventor : NOGUCHI KAZUYUKI  
SAKAMOTO KEIJI  
KIJIMA YASUSHI  
IDENO TORU  
OKITA MASAYA

## (54) OBJECT CONTROL METHOD

### (57)Abstract:

**PURPOSE:** To confirm the processing state of an object loading processing, to improve the safety and the reliability of a system, and to shorten the processing time required for change or version-up of an object.

**CONSTITUTION:** A version code showing the version of an object is given to the object having a file name for production of a version control file where the object file name and its version code are recorded. This control file is previously stored in a transmitting destination. When the object is rewritten, the version code of the object file name received from the transmitting destination is read out of the version control file stored in the transmitting destination. Then the propriety of the rewriting processing is decided from the change or no change of both version codes. In such a constitution of an object control method, if the change of the version code is decided, the rewriting processing is carried out so that an old object is rewritten into a new object. If the change of the code version is not decided, the object is not rewritten. Thus the processing time is shortened in the object control method.



## LEGAL STATUS

[Date of request for examination] 16.03.1999  
[Date of sending the examiner's decision of rejection]  
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]  
[Date of final disposal for application]  
[Patent number]  
[Date of registration]  
[Number of appeal against examiner's decision of rejection]  
[Date of requesting appeal against examiner's decision of rejection]  
[Date of extinction of right]

(19)日本国特許庁(JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-202923

(43)公開日 平成6年(1994)7月22日

(51) Int.Cl.<sup>5</sup>

G O 6 F 12/00

識別記号

5 1 7

片内整理番号

**8526-5B**

F I

### 技術表示箇所

審査請求 未請求 請求項の数 3 (全 10 頁)

(21)出願番号 特願平4-360787

(22)出願日 平成4年(1992)12月29日

(71)出願人 000002945

オムロン株式会社

京都府京都市右京区花園土堂町10番地

(72)発明者 野口 和幸

京都府京都市右京区花園土堂町10番地 才

ムロン株式会社内

(72)発明者 坂本 啓司

京都府京都市右京区花園土堂町10番地 才

ムロン株式会社内

(72)発明者 木嶋 裕史

京都府京都市右京区花園土堂町10番地 才

ムロン株式会社内

(74)代理人 弁理士 永田 良昭

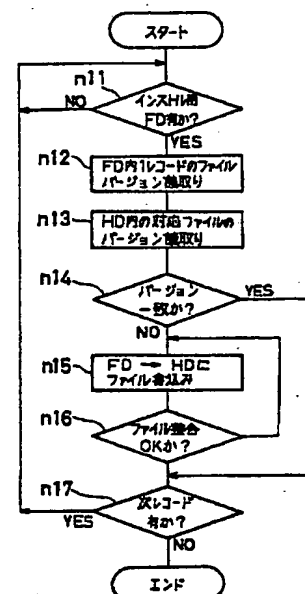
[最終頁に続く](#)

(54)【発明の名称】 オブジェクト管理方法

(57) 【要約】

【目的】この発明は、オブジェクトローディング処理では処理状況を確認することができ、システムの安全性・信頼性を向上させることができ、さらに、オブジェクトが変更やバージョンアップされた時、その処理時間を短縮することができるオブジェクト管理方法の提供を目的とする。

【構成】 ファイル名を持つオブジェクトに対して、該オブジェクトのバージョンを示すバージョン符号を付与し、オブジェクトのファイル名とそのバージョン符号とを記録したバージョン管理ファイルを作成して、予め伝送先に記憶し、上記オブジェクトの書替え時に、伝送元から伝送されたオブジェクトのファイル名のバージョン符号を、伝送先が記憶するバージョン管理ファイルから読出し、両者のバージョン符号の変化の有無を判定して書替え処理の可否判断をするオブジェクト管理方法であって、前記バージョン符号の変化の有無の判定で、変化が判定された時は書替え処理を実行して、旧オブジェクトを新オブジェクトに書替え、変化が判定されない時はオブジェクトの書替えをしないで処理時間を短縮する。



## 【特許請求の範囲】

【請求項1】ファイル名を持つオブジェクトに対して、該オブジェクトのバージョンを示すバージョン符号を付与し、オブジェクトのファイル名とそのバージョン符号とを記録したバージョン管理ファイルを作成して、予め伝送先に記憶し、上記オブジェクトの書替え時に、伝送元から伝送されたオブジェクトのファイル名のバージョン符号を、伝送先が記憶するバージョン管理ファイルから読出し、両者のバージョン符号の変化の有無を判定して書替え処理の可否判断をするオブジェクト管理方法。

【請求項2】前記バージョン符号の変化の有無の判定で、変化が判定された時は書替え処理を実行して、旧オブジェクトを新オブジェクトに書替え、変化が判定されない時はオブジェクトの書替えをしない請求項1記載のオブジェクト管理方法。

【請求項3】ファイル名を持つオブジェクトに対して、該オブジェクトのバージョンを示すバージョン符号を付与し、上記オブジェクトの書替え時に、オブジェクトのファイル名とそのバージョン符号とを記録したバージョン管理ファイルを伝送元から伝送先に伝送し、伝送先は既に記憶している旧バージョン管理ファイルと新たなバージョン管理ファイルとを比較して、バージョン符号の変化したファイル名のリストを伝送元に伝送し、伝送元はリストアップされたファイル名の新オブジェクトを伝送先に伝送し、伝送先は伝送された新オブジェクトに書替えるオブジェクト管理方法。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】この発明は、情報化システム機器において、オブジェクト（プログラム、データなど）を書替える（変更）時に有効なオブジェクト管理方法に関する。

## 【0002】

【従来の技術】従来、上述の情報化システム機器において、システムを稼動するプログラムやデータなどのオブジェクトは、決ったファイル名で外部記憶装置（例えば、フロッピディスクなど）に格納し、機器のシステムを起動するときは、上述の外部記憶装置から機器の主記憶装置（例えば、ハードディスクなど）にロードして実行している。

【0003】しかし、上述のロードにおいてシステム異常があるとダウンまたは暴走の危険性が生じるので、オブジェクトローディング処理の処理状況を確認すると、システムの安全性・信頼性が高くなるが、従来ではこの確認の処理方法が確立されていなかったために、確認できなかった。

【0004】また、任意のプログラムやデータを変更し、バージョンアップする際、これらのプログラムやデータの入換えはミスやモレがなく正確に行なう必要があるが、これを人為的に実行する場合、非常に時間がか

り、ダウンラインローディング処理でも、その幾つかのオブジェクトが変更やバージョンアップされても全てのオブジェクトをローディングするので、その処理に時間がかかる問題点を有している。

## 【0005】

【発明が解決しようとする課題】この発明は、オブジェクトローディング処理では処理状況を確認することができて、システムの安全性・信頼性を向上させることができ、さらに、オブジェクトが変更やバージョンアップされた時、その処理時間を短縮することができるオブジェクト管理方法の提供を目的とする。

## 【0006】

【課題を解決するための手段】この発明の請求項1記載の発明は、ファイル名を持つオブジェクトに対して、該オブジェクトのバージョンを示すバージョン符号を付与し、オブジェクトのファイル名とそのバージョン符号とを記録したバージョン管理ファイルを作成して、予め伝送先に記憶し、上記オブジェクトの書替え時に、伝送元から伝送されたオブジェクトのファイル名のバージョン符号を、伝送先が記憶するバージョン管理ファイルから読出し、両者のバージョン符号の変化の有無を判定して書替え処理の可否判断をするオブジェクト管理方法であることを特徴とする。

【0007】この発明の請求項2記載の発明は、上記請求項1記載の発明の構成と併せて、前記バージョン符号の変化の有無の判定で、変化が判定された時は書替え処理を実行して、旧オブジェクトを新オブジェクトに書替え、変化が判定されない時はオブジェクトの書替えをしないオブジェクト管理方法であることを特徴とする。

【0008】この発明の請求項3記載の発明は、ファイル名を持つオブジェクトに対して、該オブジェクトのバージョンを示すバージョン符号を付与し、上記オブジェクトの書替え時に、オブジェクトのファイル名とそのバージョン符号とを記録したバージョン管理ファイルを伝送元から伝送先に伝送し、伝送先は既に記憶している旧バージョン管理ファイルと新たなバージョン管理ファイルとを比較して、バージョン符号の変化したファイル名のリストを伝送元に伝送し、伝送元はリストアップされたファイル名の新オブジェクトを伝送先に伝送し、伝送先は伝送された新オブジェクトに書替えるオブジェクト管理方法であることを特徴とする。

## 【0009】

【発明の効果】この発明の請求項1記載の発明によれば、オブジェクトのバージョンを管理することで、例えば、オブジェクトの伝送元と伝送先とのバージョン符号が一致しないときは、トラブルが発生したと判断することができて、エラー処理ができるため、オブジェクトローディング処理ではシステム異常によりシステムのダウンや暴走の危険性をなくすることができ、システムの安全性・信頼性を向上させることができる。

【0010】この発明の請求項2記載の発明によれば、オブジェクトの変更やバージョンアップする場合、そのオブジェクトのバージョン符号を変えることにより、伝送先ではバージョン符号の変わったオブジェクトのみ書替えればよいので、処理時間が短縮される。

【0011】この発明の請求項3記載の発明によれば、ダウンラインローディング処理では全てのオブジェクトをローディングするのではなく、リストアップした変更のあるオブジェクトのみをローディングすればよく、伝送時間が短縮されて、ダウンラインローディング処理時間10が短縮される。

【0012】

【実施例】この発明の一実施例を以下図面に基づいて詳述する。

【0013】図面はオブジェクト管理方法を示し、図1において、該端末Aは情報化システム機器の一例であって、CPU10、プログラムを格納したROM11、ハードディスクのようなRAM12、フロッピーディスクのような外部記憶装置13、データの表示を行なう表示装置14、データの入力を行なうキーボード15、他の機器との通信を可能にするための通信制御部16を備えている。

【0014】そして、この端末Aを起動するには、外部記憶装置13からRAM12にオブジェクトローディングすることにより起動される。

【0015】上述のRAM12には、予めまたは前回のバージョン管理ファイルとオブジェクトファイルとを記憶しており、バージョン管理ファイルは図2に示すように、各オブジェクトにファイル名を設定し、各オブジェクトファイルにはそのバージョンを示すバージョン符号を設定して、これらオブジェクトファイル名とバージョン符号とで1レコードとしている。そして、各オブジェクトファイルにはファイル名に対応してそれぞれのオブジェクトをファイルしている。

【0016】上述のようにして構成した端末AのCPU10が処理するオブジェクトローディングを、図3を参照して説明する。

【0017】外部記憶装置13で記憶媒体から入力されるバージョン管理ファイルより1レコードを読み取り（ステップn1）、指定されたバージョンのオブジェクトファイルを予め記憶、または前回記憶したRAM12から検索する（ステップn2）。

【0018】指定されたオブジェクトファイルがあったか否かを判定し（ステップn3）、オブジェクトファイルがあった場合は、正常なオブジェクトであると判定して、指定のエリアにロードする（ステップn4）。しかし、オブジェクトファイルがない場合はオブジェクトに異常があったことになり、エラー処理を実行する（ステップn5）。このようなオブジェクトのチェックを各レコードごとに実行する（ステップn6）。

【0019】上述のようにオブジェクトファイルの有無を判定することで、エラーが生じたときは、システム起動に要するオブジェクトファイルがないことになるので、未然にシステム異常を発見することができる。

【0020】ついでオブジェクトのインストール処理について説明する。図4で示すバージョン管理ファイルはオブジェクトファイル名Aがバージョンアップした例を示し、図イは旧バージョンで、図ロが新バージョンであって、他のファイル名B、Cは旧バージョンのままである。従前であれば、全ファイル名A、B、Cを書替えるが、この実施例ではバージョンアップされたファイル名Aのみを書替えることで済む。図5に基づいてインストール処理動作を説明する。

【0021】すなわち、外部記憶装置13で記憶媒体（例えば、フロッピーディスクなど）から入力されるバージョン管理ファイルより1レコードを読み取り（ステップn11、n12）、指定されたバージョンのオブジェクトファイルを、予め、または前回記憶したRAM12から検索する（ステップn13）。指定されたオブジェクトファイルからバージョン符号を読み取って、新たに入力されたファイルのバージョン符号と比較して、一致するか否かを判定する（ステップn14）。

【0022】前述の図4のオブジェクトファイル名Aのようにバージョンアップされている場合は、バージョン符号が不一致であるため、バージョンアップが判定され、このオブジェクトファイル名Aを指定のエリアに記憶し（ステップn15）、さらに、ファイルのサム値を比較して、ファイルの整合をチェックし（ステップn16）、その整合が確認できれば、次のレコードの処理を実行する（ステップn17）。

【0023】しかし、図4のオブジェクトファイル名B、Cのようにバージョンが変わらない場合は、バージョン符号が一致するため、既に記憶しているものと同一のオブジェクトファイルと判定し、ファイルの書替えをしないで次の処理に移行する。

【0024】上述のように、バージョンが同一のファイルは書替えをせずにバージョンアップされたファイルのみを書替えることで、処理時間を短縮することができる。

【0025】ついでダウンラインローディング処理について説明する。図6において、伝送元の端末Aから伝送先の端末Bに伝送線20を介してオブジェクトをダウンラインローディング処理（DLL）する場合を示し、端末A、Bは共に、前述の図1で示した構成を有している。

【0026】図7～図9を参照して、端末Aから端末Bに対するダウンラインローディング処理（DLL）を説明する。端末Bには予めまたは前回のバージョン管理ファイルおよび各ファイル名のオブジェクトファイルを持っており（ステップn21）、上述のバージョン管理フ

ファイルは前述の図2で示したファイルと等価であって、各オブジェクトにはファイル名を設定し、各オブジェクトにはそのバージョンを示すバージョン符号を設定して、これらオブジェクトファイル名とバージョン符号とで1レコードとしている。

【0027】伝送元である端末Aからは新たなバージョン管理ファイルを伝送先である端末Bに伝送し（ステップn22）、端末Bはそれを記憶する（ステップn23）。端末Bでは上述の新バージョン管理ファイルから1レコードを読み出し（ステップn24）、指定されたバージョンのオブジェクトファイルを検索する（ステップn25）。

【0028】指定したオブジェクトファイル名とそのバージョン符号が一致するオブジェクトファイルがある場合（ステップn26のYES判定時）は、バージョンの変更がないため次のステップに移行する。しかし、一致するオブジェクトファイルが見つからない場合（ステップn26のNO判定時）は、端末Bでは不足するオブジェクトファイルであるため、該ファイル名を図10に示すように、不足オブジェクトファイルにリストアップする（ステップn27）。

【0029】ついで次のレコードがあれば（ステップn28のYES判定時）、前述のステップn24にリターンするが、次のレコードがない時（ステップn28のNO判定時）、不足オブジェクトファイルにリストアップしたレコード数を計数して、該レコード数が零の時（ステップn29のYES判定時）は、端末AにDLL終了通知を伝送する（ステップn30）して、処理を終了する。

【0030】しかし、リストアップしたレコード数がある時（ステップn29のNOの判定時）は、リストアップした不足オブジェクトファイルのリストデータを端末Aに伝送する（ステップn31）。

【0031】端末Aは端末Bからの伝送を待っており（ステップn32）、端末Bから伝送があると、DLL終了通知か否かを判定し（ステップn33）、DLL終了通知であれば（同ステップのYES判定時）処理を終了する。

【0032】しかし、不足オブジェクトファイルのリストアップの伝送であれば（ステップn33のNO判定時）、該リストを記憶し（ステップn34）、該リストから不足オブジェクトファイルの1レコードを読み出して（ステップn35）、端末Aで保有しているオブジェクトファイルより上述の不足オブジェクトファイルを検索する（ステップn36）。

【0033】上述の検索で不足のオブジェクトファイルが見つからなかった時（ステップn37のNO判定時）は、先のステップn22で伝送したデータがエラーかまたは端末Bでのダウンラインローディング処理がエラーか、何れにせよ今回のダウンラインローディング処理が

エラーであるため、端末Bにダウンラインローディング処理取消し指示を伝送する（ステップn38）。

【0034】一方、端末Bの不足オブジェクトファイルが見つかった場合（ステップn37のYES判定時）は、該不足オブジェクトファイルを端末Bに伝送する（ステップn39）。その後、次の不足オブジェクトファイルリストに次のレコードがあれば（ステップn40のYESの判定時）、ステップn35にリターンし、該レコードが終了していれば（ステップn40のNO判定時）、端末BにDLL完了通知を伝送する（ステップn41）。

【0035】端末Bは端末Aからの伝送を待っており（ステップn42）、端末Aから伝送があると、DLL取消し通知か否かを判定し（ステップn43）、DLL取消し通知であれば（同ステップのYES判定時）、DLL取消し処理を実行する（ステップn44）。すなわち、今回書き込んだオブジェクトファイルの全てを削除し、今回の新バージョン管理ファイルを削除する。

【0036】前述のステップn43の判定でNOの判定であれば、続いてDLL完了通知であるか否かを判定し（ステップn45）、DLL完了通知であれば（同ステップのNO判定時）、DLL完了処理を実行する（ステップn46）。すなわち、今回書き込んだ全てのオブジェクトファイルを本場オブジェクトファイルに書替え、さらに、今回の新バージョン管理ファイルを本場バージョン管理ファイルに書替える。

【0037】前述のステップn45の判定でNOの判定であれば、端末Aから伝送されたデータが不足オブジェクトファイルであるため、これを外部記憶装置13に記憶してステップn42にリターンし（ステップn47）、前述のステップn42、n43、n45、n47の処理を繰返すことで、全部の不足オブジェクトファイルを受取ることができる。そして、端末Bが不足オブジェクトファイルの全部を受取ると、その最終では端末AからDLL完了通知が伝送される。

【0038】なお、上述の不足オブジェクトファイル（または差分情報）を外部記憶装置13に記憶しておくと、RAM12のハードディスクが故障したときのバックアップになり、RAM12のハードディスク修理後に、外部記憶装置13のオリジナルフロッピーディスクからのオブジェクトファイルと、上述の不足オブジェクトファイル（または差分情報）で端末Bを起動するためのオブジェクトファイルを復旧することができる。

【0039】前述のステップn44のDLL取消し処理またはステップn46のDLL完了処理が終わると、端末Bは端末Aに対しDLL終了通知を伝送し（ステップn48）、端末Aはその終了通知を受取ること（ステップn49）、ダウンラインローディング処理を終了する。上述の実施例によれば、不足のオブジェクトファイルのみを伝送するようにしているので、ダウンラインロ

ーディング処理の伝送時間を短縮し、その処理時間を短縮することができる。

【図面の簡単な説明】

【図1】 端末の制御回路ブロック図。

【図2】 バージョン管理ファイルの説明図。

【図3】 オブジェクトローディングのフローチャート。

【図4】 バージョンアップしたバージョン管理ファイルの説明図。

【図5】 インストール処理のフローチャート。

【図6】 端末Aから端末Bへの伝送の構成図。

【図7】 ダウンラインローディング処理のフローチャー

ト。

【図8】 図7の続きを示すフローチャート。

【図9】 図8の続きを示すフローチャート。

【図10】 不足オブジェクトファイルのリストアップを示す説明図。

【符号の説明】

A, B…端末

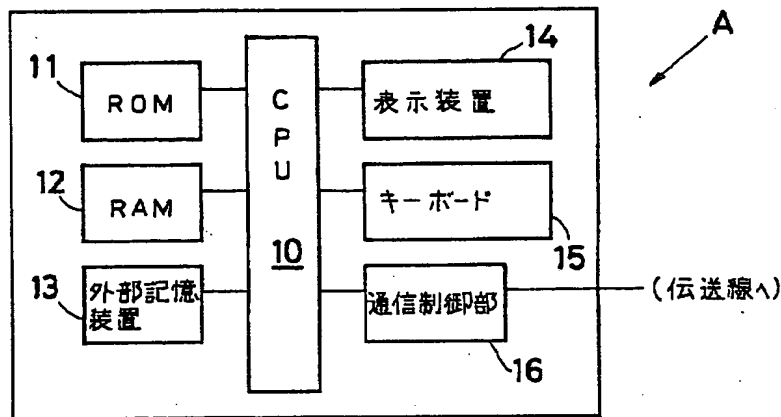
10…CPU

11…ROM

12…RAM

13…外部記憶装置

【図1】



【図2】

端末Aバージョン管理ファイル	
オブジェクトファイル名1	バージョン符合
オブジェクトファイル名2	バージョン符合
⋮	⋮
⋮	⋮

1レコード

【図4】

端末Aの旧バージョン管理ファイル	
オブジェクトファイル名 A	バージョン 1.1
オブジェクトファイル名 B	バージョン 1.1
オブジェクトファイル名 C	バージョン 1.1

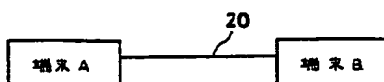
(イ)



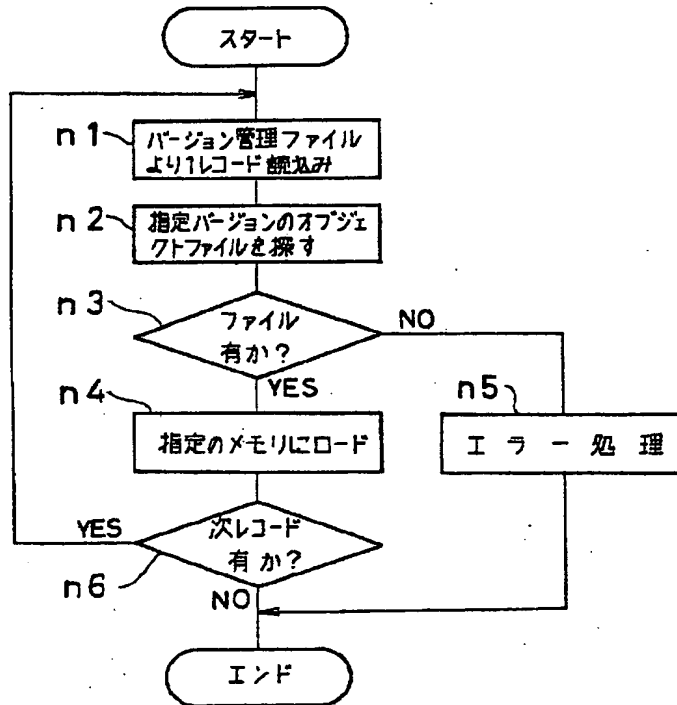
新バージョン管理ファイル	
オブジェクトファイル名 A	バージョン 1.5
オブジェクトファイル名 B	バージョン 1.1
オブジェクトファイル名 C	バージョン 1.1

(ロ)

【図6】



【図3】

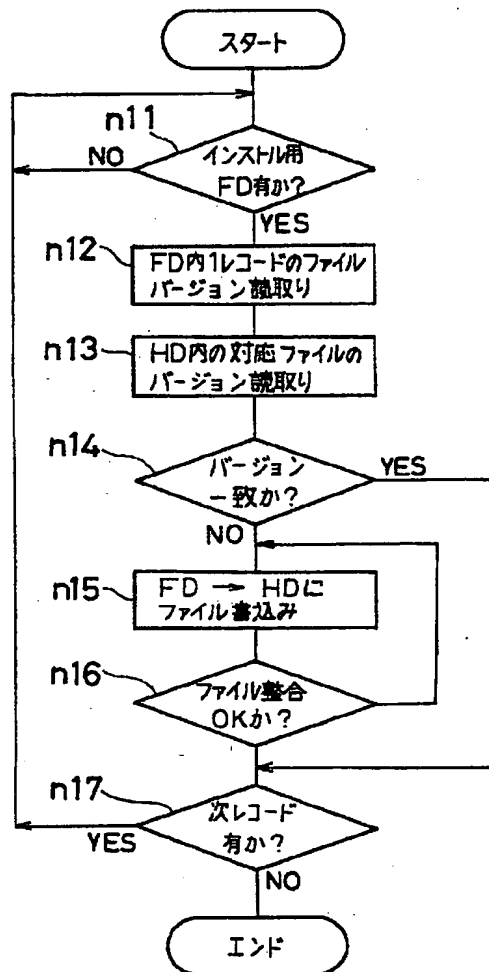


【図10】

検索日不足オブジェクトファイルリスト

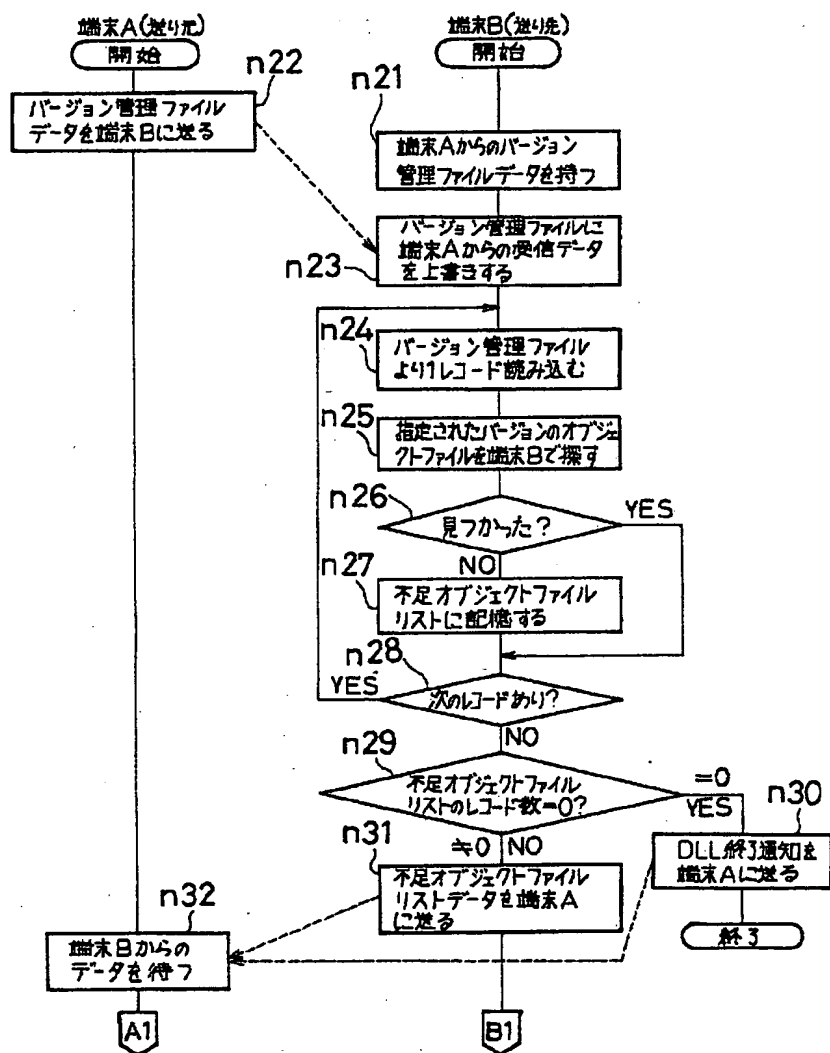
不足オブジェクトファイル名1	バージョン符合
不足オブジェクトファイル名2	バージョン符合
⋮	⋮
⋮	⋮
⋮	⋮

【図5】

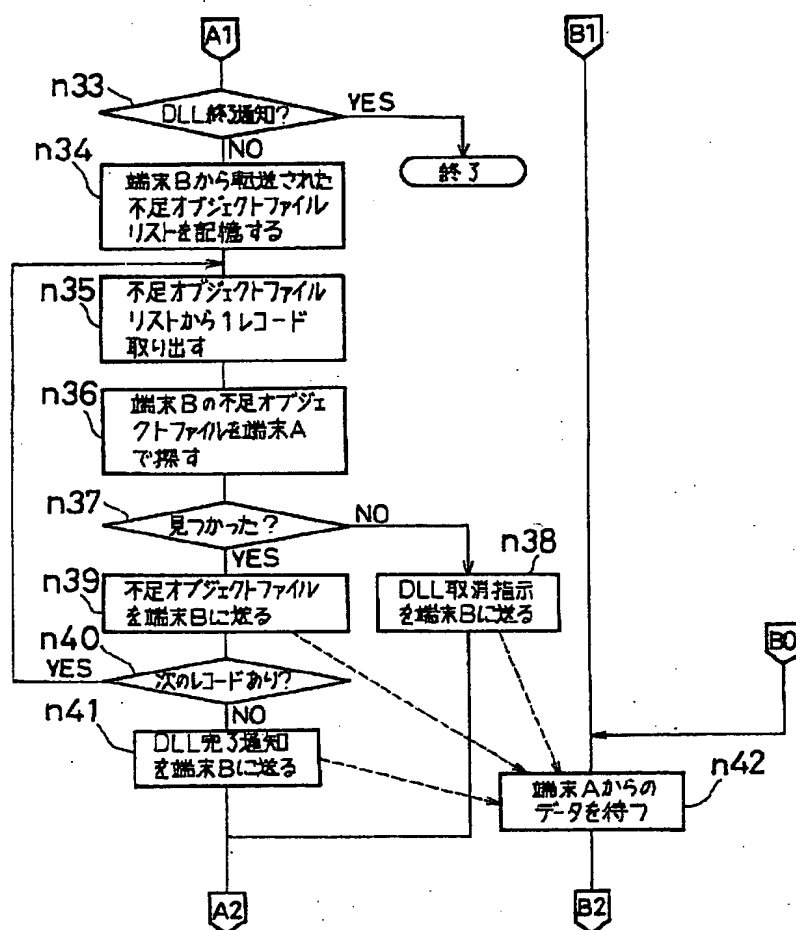




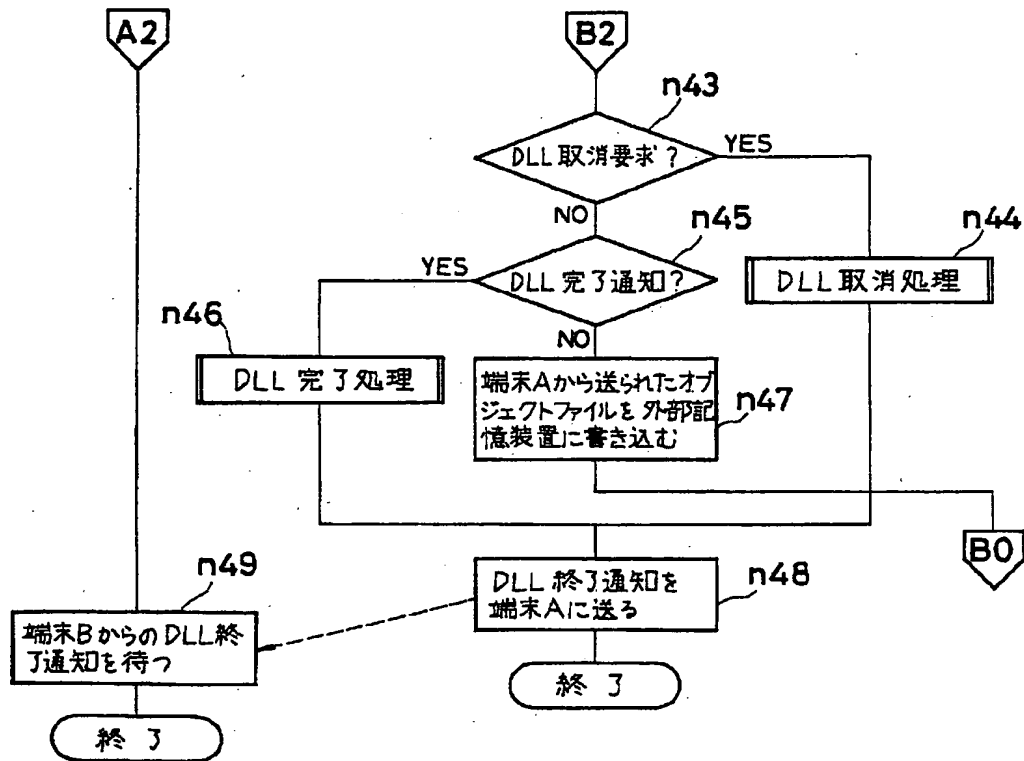
【図7】



【図8】



【図9】



フロントページの続き

(72)発明者 出野 徹  
 京都府京都市右京区花園土堂町10番地 オ  
 ムロン株式会社内

(72)発明者 沖田 昌也  
 京都府京都市右京区花園土堂町10番地 オ  
 ムロン株式会社内